# Лабораторна робота №6.

**Мета:**

● Ознайомлення з моделлю потоків ​ Java ​ .

● Організація паралельного виконання декількох частин програми.

● Вимірювання часу паралельних та послідовних обчислень.

● Демонстрація ефективності паралельної обробки.

# Вимоги

1. Використовуючи програми рішень попередніх задач, продемонструвати можливість паралельної обробки елементів контейнера: створити не менше трьох додаткових потоків, на яких викликати відповідні методи обробки контейнера.

2. Забезпечити можливість встановлення користувачем максимального часу виконання (таймаута) при закінченні якого обробка повинна припинятися незалежно від того знайдений кінцевий результат чи ні.

3. Для паралельної обробки використовувати алгоритми, що не змінюють початкову колекцію.

4. Кількість елементів контейнера повинна бути досить велика, складність алгоритмів обробки колекції повинна бути зіставна, а час виконання приблизно однаковий, наприклад:

○ пошук мінімуму або максимуму;

○ обчислення середнього значення або суми;

○ підрахунок елементів, що задовольняють деякій умові;

○ відбір за заданим критерієм;

○ власний варіант, що відповідає обраній прикладної області.

5. Забезпечити вимірювання часу паралельної обробки елементів контейнера за допомогою розроблених раніше методів.

6. Додати до алгоритмів штучну затримку виконання для кожної ітерації циклів поелементної обробки контейнерів, щоб загальний час обробки був декілька секунд.

7. Реалізувати послідовну обробку контейнера за допомогою методів, що використовувались для паралельної обробки та забезпечити вимірювання часу їх роботи.

8. Порівняти час паралельної і послідовної обробки та зробити висновки про ефективність розпаралелювання: ○ результати вимірювання часу звести в таблицю; ○ обчислити та продемонструвати у скільки разів паралельне виконання швидше послідовного.

# Розробник

Горностай Богдан, КН-108

# Задача

8. Автостанція. Запис в розкладі: номер рейсу; час відправлення; дні тижня; кількість вільних місць; маршрут - необмежений набір значень у вигляді “назва станції, час прибуття”.

# Опис програми

Програма працює з багатопотоковістю, перевіряє та порівнює роботи з увімкнутим ядром та без нього за допомогою простих алгоритмів, які не змінюють початкову колекцію.

# Засоби ООП

Були використані різні класи та методи, структури даних та модифікатори доступу також бібліотека Thread

# Ієрархія та структура класів

# [Main](https://github.com/BohdanHornostai/OOP/blob/master/Lab3/Main.java) – реалізує меню програми, викликає процеси в різних потоках, десереалізує файли.

# [AutoStation](https://github.com/BohdanHornostai/OOP/blob/master/Lab3/AutoStation.java) – domain-об’єкт

# 2.4 Варіанти використання

Можна використовувати для прикладу швидкодії роботи з ядром та без нього.

# ВИСНОВКИ

Ми ознайомились з моделлю потоків в Java, організували паралельне виконання частин програми, виміряли час паралельних та послідовних обчислень та продемонстрували ефективність паралельної обробки.